



Material de Apoio para Estudo

Teste de Transição de Estado

Versão 1.2

Direitos Autorais

Copyright© Brazilian Software Testing Qualifications Board (doravante denominado BSTQB®)

BSTQB® é uma marca registrada da ABRAMTI Associação Brasileira de Melhoria em TI.

BSTQB® é o Conselho Membro do ISTQB® International Software Testing Qualifications Board representando o Brasil nesta instituição.

Copyright©2023 autores da versão inicial (ordem alfabética): George Fialkovitz, Irene Nagase, Osmar Higashi e Stênio Viveiros.

Todos os direitos reservados. Os autores transferem os direitos autorais para o *Brazilian Software Testing Qualifications Board* (BSTQB®). Os autores (como detentores atuais de direitos autorais) e o BSTQB® (como futuro detentor dos direitos autorais) concordaram com as seguintes condições de uso:

- Este material foi produzido para apoiar o estudo do candidato interessado nos exames de certificação em Teste de Software do ISTQB®.
- Este material não pode ser comercializado.
- Extratos deste documento podem ser copiados se a fonte for reconhecida.
- Qualquer indivíduo ou grupo de indivíduos pode usar este material como base para artigos e livros, se os autores e o BSTQB® forem reconhecidos como a fonte e os proprietários dos direitos autorais

Histórico

Versão	Data	Observação
0.0	02/08/2023	Versão inicial
0.1	22/08/2023	Inclusão de exemplo de teste de transição de estado
1.0	30/08/2023	Lançamento
1.1	05/03/2024	Correção de descritivo da fórmula de transição de estado
1.2	28/06/2024	Adequação visual

Sumário

Direitos Autorais	2
Histórico	3
Sumário	3
1 Introdução.....	4
1.1 O que é Diagrama de Transição de Estado	4
2 Teste de Transição de Estado.....	5
2.1 Benefícios	5
2.2 Dificuldades.....	6
2.3 Exemplo de Teste de Transição de Estado	7
3 Cobertura do Teste de Transição de Estado	9
3.1.1 Exemplo de Cobertura de Teste de Transição de Estado.....	10
4 Recapitulando.....	11
5 Referências.....	12

1 Introdução

O Teste de Transição de Estado é uma técnica de Teste Caixa-Preta que examina o comportamento do sistema durante as mudanças de estado. Estados podem ser definidos como condições ou situações em que o sistema se encontra em momentos específicos da sua execução. O teste de transição de estado ajuda a validar a lógica do sistema e sua resposta às ações ou eventos que causam mudanças de estado.

1.1 O que é Diagrama de Transição de Estado

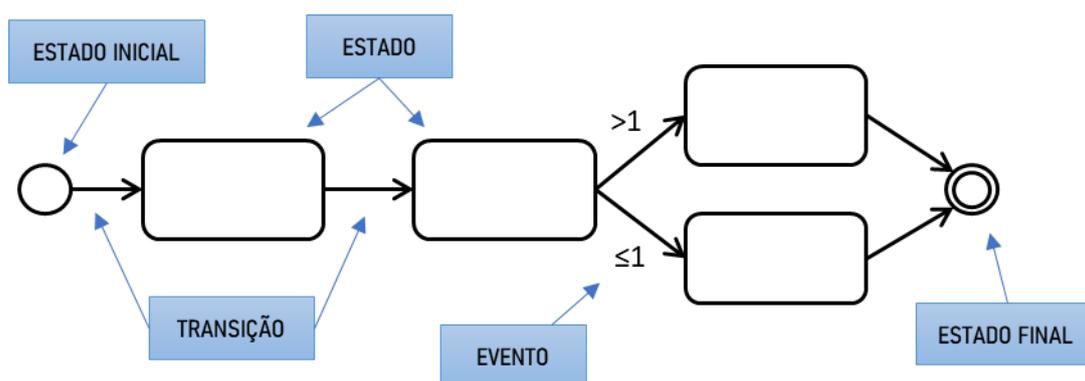
Um diagrama de transição de estado, também conhecido como máquina de estado ou diagrama de estados, é uma representação gráfica que descreve os possíveis estados de um sistema ou objeto e as transições entre esses estados em resposta a eventos específicos.

Esses diagramas são amplamente utilizados em modelagem de sistemas complexos, como software, sistemas embarcados, máquinas, entre outros, para visualizar e entender o comportamento do sistema em diferentes situações.

Os diagramas de transição de estado são úteis para modelar e compreender o comportamento complexo de sistemas que possuem estados e transições bem definidos. Eles são frequentemente utilizados durante o processo de análise e projeto de sistemas para identificar casos de uso, cenários e requisitos de software.

Além disso, esses diagramas podem ser uma ferramenta valiosa para comunicar de forma clara e eficaz o comportamento de um sistema para os membros da equipe de desenvolvimento, stakeholders e outros envolvidos no projeto.

Principais elementos de um diagrama de transição de estado:



- **Estados Inicial:** Ponto de entrada do diagrama de estados.
- **Estado:** Representa uma condição específica em que o sistema ou objeto se encontra.
- **Transição:** Indica uma mudança de estado causada por um evento específico.
- **Evento:** Representa uma ação ou ocorrência que desencadeia uma transição de estado.
- **Estados Final:** Ponto de saída do diagrama de estados.

2 Teste de Transição de Estado

O Teste de Transição de Estado é uma técnica essencial para garantir que o software responda adequadamente às mudanças de estado e que a lógica do sistema esteja correta. Ao testar as transições entre estados, podemos melhorar a qualidade e a confiabilidade do produto final, garantindo uma experiência positiva para os usuários.

É importante garantir que o software reaja corretamente e de forma consistente quando ocorrerem eventos de transição. O Teste de Transição de Estado ajuda a eliminar problemas e defeitos antes do lançamento do software, melhorando a qualidade do produto.

2.1 Benefícios

O teste de transição de estados oferece diversos benefícios significativos para o processo de desenvolvimento e garantia da qualidade do software. Alguns dos principais benefícios são:

Identificação de Bugs de Lógica: O teste de transição de estados ajuda a identificar bugs e problemas de lógica no software. Ao verificar como o sistema responde às transições entre estados, é possível detectar comportamentos inesperados ou não planejados.

Validação das Mudanças de Estado: Garante que as mudanças de estado do sistema ocorram corretamente e de forma consistente, sem causar efeitos colaterais indesejados.

Melhoria da Robustez: Ao testar as transições de estado, é possível garantir que o software seja robusto e capaz de lidar com diferentes cenários e sequências de eventos.

Verificação de Fluxos de Trabalho: O teste de transição de estados permite verificar se os fluxos de trabalho e as funcionalidades essenciais do sistema estão sendo executados corretamente.

Detecção de Deadlocks e Condições de Corrida: É possível identificar problemas como deadlocks e condições de corrida, que podem ocorrer quando há transições concorrentes de estados.

Validação da Experiência do Usuário: Testar as transições de estado também ajuda a garantir que a experiência do usuário seja suave e que as ações e eventos do sistema sejam refletidos corretamente na interface do usuário.

Redução de Erros em Produção: Ao eliminar bugs e falhas durante o teste de transição de estados, a probabilidade de erros em produção é reduzida, evitando retrabalho e custos adicionais.

Garantia de Conformidade: É possível verificar se o software está em conformidade com os requisitos de negócio, regulamentações e padrões.

Melhoria da Qualidade do Software: Através do teste de transição de estados, a qualidade geral do software é aprimorada, proporcionando um produto mais confiável e estável para os usuários finais.

Maior Satisfação do Cliente: A identificação precoce e a correção de problemas relacionados às transições de estado resultam em um produto final mais confiável e com menor probabilidade de causar frustração aos usuários, aumentando a satisfação do cliente.

Em resumo, o teste de transição de estados é uma etapa crucial do processo de teste de software, pois garante que o sistema funcione corretamente durante mudanças de estado e transições entre diferentes condições. Através desses testes, é possível melhorar a qualidade do produto final e a experiência do usuário, reduzindo o risco de falhas e problemas em produção.

2.2 Dificuldades

Os testes de transição de estado podem apresentar diversos problemas e desafios que devem ser considerados ao planejar e executar essa atividade de teste. Alguns deles são:

Complexidade do sistema: Sistemas com múltiplos estados e muitas transições podem ser complexos para testar de forma abrangente. Identificar todos os possíveis cenários de transição pode ser um desafio, especialmente em sistemas com muitos estados interconectados.

Cobertura de teste insuficiente: Garantir que todos os caminhos de transição de estado sejam testados pode ser difícil. A cobertura insuficiente pode levar à falta de detecção de problemas de lógica ou a não identificação de comportamentos inesperados.

Interdependência de transições: Em sistemas com várias transições concorrentes ou interdependentes, a sequência exata dos eventos pode ser difícil de reproduzir durante os testes, tornando difícil identificar e reproduzir erros.

Tempo e recursos limitados: Testar todas as combinações possíveis de estados e eventos pode ser inviável em termos de tempo e recursos. A seleção adequada de cenários de teste é fundamental para otimizar a eficiência do teste.

Ambiente de teste adequado: Em alguns casos, pode ser difícil reproduzir todas as condições necessárias para realizar testes de transição de estado em um ambiente de teste controlado.

Mudanças na lógica do sistema: À medida que o sistema evolui e é atualizado, novas transições de estado podem ser adicionadas ou lógicas existentes podem ser alteradas, o que exige a atualização contínua dos testes.

Documentação deficiente: Se a documentação dos estados, transições e condições for incompleta ou imprecisa, os testes de transição de estado podem ser prejudicados.

Testes de regressão: Conforme o software é atualizado, é necessário realizar testes de regressão nos testes de transição de estado para garantir que as mudanças não impactem negativamente as transições existentes.

Captura de dados de teste: Em alguns cenários de transição de estado, pode ser difícil capturar e verificar os dados gerados durante o teste.

Para enfrentar esses problemas e desafios, é importante ter uma abordagem estruturada de teste e priorizar os cenários de teste mais críticos e representativos. O uso de ferramentas de automação de teste, criação de matrizes de transição de estado e adoção de técnicas como Teste de Tabela de Decisão podem ajudar a aumentar a eficiência e eficácia dos testes de transição de

estado. Além disso, a colaboração entre desenvolvedores e testadores é fundamental para garantir uma cobertura abrangente e identificar problemas de forma proativa.

2.3 Exemplo de Teste de Transição de Estado

Cenário: Transições de Estado em um Sistema Bancário

CC

Conta Criada: Uma nova conta foi criada, mas ainda não possui saldo.

CA

Conta Ativa: A conta possui um saldo positivo e está pronta para ser usada.

CB

Conta Bloqueada: A conta possui um saldo negativo ou algum problema, e foi bloqueada para transações.

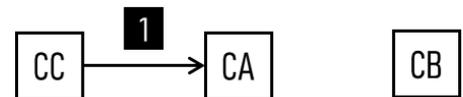
Agora, vamos olhar para os exemplos de testes de transição de estado:

(1) Transição da Conta Criada para a Conta Ativa:

Pré-condição: Conta Criada.

Ação: Fazer um depósito inicial de um valor positivo.

Pós-condição esperada: A conta deve estar no estado Conta Ativa.

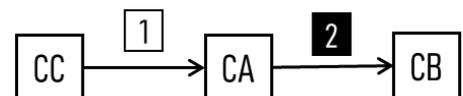


(2) Transição da Conta Ativa para a Conta Bloqueada:

Pré-condição: Conta Ativa com saldo positivo.

Ação: Realizar saques que resultam em um saldo negativo.

Pós-condição esperada: A conta deve estar no estado Conta Bloqueada.

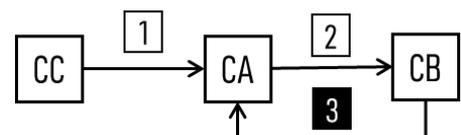


(3) Transição da Conta Bloqueada para a Conta Ativa:

Pré-condição: Conta Bloqueada.

Ação: Fazer um depósito que resulta em um saldo positivo.

Pós-condição esperada: A conta deve estar no estado Conta Ativa.

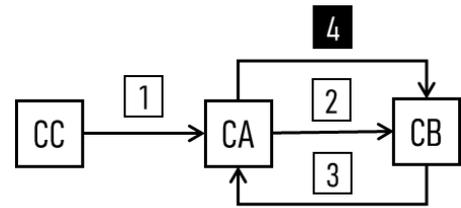


(4) Transição da Conta Ativa para a Conta Bloqueada (por razões externas):

Pré-condição: Conta Ativa com saldo positivo.

Ação: Receber uma notificação externa para bloquear a conta.

Pós-condição esperada: A conta deve estar no estado Conta Bloqueada.

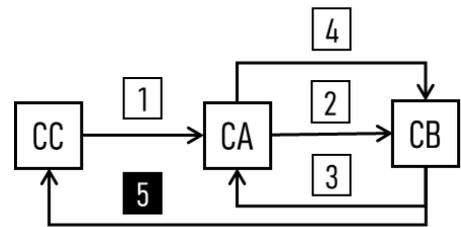


(5) Transição da Conta Bloqueada para a Conta Criada (por correção de problemas):

Pré-condição: Conta Bloqueada.

Ação: Corrigir os problemas que levaram ao bloqueio.

Pós-condição esperada: A conta deve estar no estado Conta Criada.



Este é apenas um exemplo de teste de transição de estado em um sistema bancário. Cada teste aborda uma transição específica entre os estados das contas, garantindo que o sistema se comporte corretamente em diferentes situações. Lembre-se de que é importante testar tanto os cenários esperados quanto os cenários de exceção para garantir a robustez e a confiabilidade do sistema bancário.

3 Cobertura do Teste de Transição de Estado

Existem diversos critérios de cobertura para o teste de transição de estado, e este syllabus aborda três deles.

O primeiro critério é a **cobertura de todos os estados**, onde os itens de cobertura são os próprios estados do sistema. Para alcançar uma cobertura de 100% de todos os estados, os casos de teste devem garantir que cada estado seja visitado. Essa cobertura é medida pelo percentual de estados visitados pelo número total de estados.

$$\frac{\textit{Estados Executados}}{\textit{Total de Estados}}$$

O segundo critério é a **cobertura de transições válidas**, também conhecida como cobertura de 0-switch. Neste caso, os itens de cobertura são as transições válidas únicas do sistema. Para atingir 100% de cobertura de transições válidas, os casos de teste devem executar todas as transições válidas possíveis. A cobertura é o percentual de transições válidas executadas pelo número total de transições válidas.

$$\frac{\textit{Transições Válidas Executadas}}{\textit{Total de Transições Válidas}}$$

O terceiro critério é a **cobertura de todas as transições**, onde os itens de cobertura são todas as transições apresentadas em uma tabela de estados. Para alcançar 100% de cobertura, os casos de teste devem executar todas as transições válidas e, idealmente, tentar executar as transições inválidas também. Incluir testes com transições inválidas ajuda a evitar o mascaramento de falhas, onde um defeito impede a detecção de outro. A cobertura é medida pelo número de transições válidas e inválidas executadas ou tentadas, dividido pelo número total de transições válidas e inválidas, também expressa em forma de porcentagem.

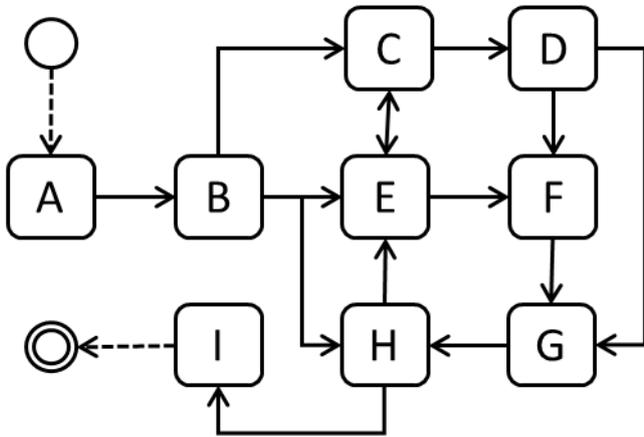
$$\frac{\textit{Transições Válidas Executadas} + \textit{Transições Inválidas Executadas}}{\textit{Total de Transições Válidas} + \textit{Total de Transições Inválidas}}$$

Comparando os critérios, a cobertura de todos os estados é considerada mais fraca em relação à cobertura de transições válidas, pois pode ser alcançada sem executar todas as transições possíveis. A cobertura de transições válidas é o critério de cobertura mais amplamente utilizado, pois garante tanto a cobertura total de todos os estados quanto a cobertura total de transições válidas. Em sistemas de missão e segurança crítica, atingir a cobertura total de todas as transições é um requisito mínimo recomendado.

Dessa forma, é essencial selecionar o critério de cobertura adequado ao contexto do sistema em teste, garantindo que os casos de teste sejam eficientes e eficazes para identificar possíveis falhas e garantir a qualidade do software.

3.1.1 Exemplo de Cobertura de Teste de Transição de Estado

Diagrama de Estados foi submetido aos seguintes casos de teste.



Casos de Teste	
CT1	CT2
A	A
B	B
C	H
D	I
G	
H	
I	

Qual a cobertura atingida após a execução dos dois casos de teste?

Resolução:

Existe um total de 14 transições de estado identificadas no Diagrama de Estado: AB, BC, CD, CE, EC, DF, BE, BH, EF, DG, HE, FG, GH e HI

O caso de teste CT1 executou 6 transições de estado: AB, BC, CD, DG, GH e HI.

O caso de teste CT2 executou 3 transições de estado: AB, BH e HI.

Mas as transições de estado executadas em CT1, AB e HI, já foram executadas pelo caso CT1, assim, apenas a BH foi uma nova transição executada neste caso de teste.

Para determinar a cobertura das transições executadas, usamos a fórmula:

$$\frac{\text{Transições de Estado Executadas}}{\text{Total de Transições de Estado}}$$

Logo:

$$\text{Cobertura} = \frac{6 + 1}{14} = \frac{7}{14} = \frac{1}{2} = 0,5$$

A cobertura foi de **50%**

4 Recapitulando...

O Teste de Transição de Estado, também conhecido como Teste de Estado ou Teste de Máquina de Estado, é uma técnica de Teste Caixa-Preta que se concentra em verificar o comportamento de um sistema ou programa em diferentes estados e suas transições. É especialmente útil para sistemas que possuem lógica complexa e interações sequenciais entre os estados.

O objetivo principal desse tipo de teste é garantir que todas as transições possíveis entre estados sejam adequadamente testadas e que o software responda corretamente a essas mudanças de estado. Isso ajuda a identificar problemas de funcionamento, comportamento inesperado ou erros de programação que podem ocorrer quando o sistema muda de um estado para outro.

O Teste de Transição de Estado é particularmente útil para sistemas com lógica complexa, como controladores de dispositivos, sistemas embarcados, protocolos de comunicação, entre outros. Ele ajuda a garantir que o software funcione corretamente em diferentes cenários e condições de operação, tornando-se uma parte importante do processo geral de garantia de qualidade do software.

5 Referências

ISTQB Syllabus CTFL

Certified Tester Foundation Level, v4.0, bstqb.org.br

Copeland, L. (2004)

A Practitioner's Guide to Software Test Design, Artech House: Norwood MA

Jorgensen, P. (2014)

Software Testing, A Craftsman's Approach (4e), CRC Press: Boca Raton FL

TMap

State Transition Testing, <https://www.tmap.net/wiki/state-transition-testing>